**EDA Tool Developer 실습 프로젝트 보고서**

**1. 개요**

본 프로젝트는 RTL(Register Transfer Level)부터 GDSII까지의 전체 반도체 설계 플로우를 실습하고, 오픈소스 EDA 툴의 분석과 커스터마이징을 통해 반도체 EDA Tool 개발자로서의 기반을 다지는 것을 목표로 하였다. 실습은 2025년 4월 22일부터 5월 23일까지 약 4주간 진행되었으며, 주말과 공휴일을 제외한 평일 중심으로 구성되었다.

**2. 목표**

* 오픈소스 EDA 툴(Yosys, OpenROAD 등)의 설치 및 사용법 숙지
* 간단한 Verilog 설계를 RTL-to-GDSII 플로우로 구현
* 각 EDA 툴의 동작 방식 분석 및 .tcl 스크립트 이해
* EDA Tool 사용자 → 개발자 관점 전환 실습
* GitHub 기반 문서화 및 포트폴리오 정리

**3. 실습 환경**

* OS: Ubuntu 22.04 (WSL2 환경)
* 사용 툴:
  + Yosys: RTL 합성(Synthesis)
  + OpenROAD: Floorplanning, Placement, Routing, CTS, GDS Export
  + Icarus Verilog + GTKWave: RTL 시뮬레이션
* PDK: Sky130 (Google/SkyWater 제공 오픈소스 PDK)
* 문서화: Markdown 기반 정리 (GitHub 업로드 예정)

**4. 주간 진행 내용**

Week 1 : 환경 구축 및 간단한 RTL 테스트

* Yosys, OpenROAD, GTKWave 설치 및 테스트
* 간단한 Verilog 회로(and\_gate, full\_adder) 설계 및 시뮬레이션
* Yosys를 통한 합성 → OpenROAD에 netlist 전달 테스트
* OpenROAD flow에 필요한 필수 .tcl 스크립트 구조 파악  
  → init\_floorplan.tcl, helpers.tcl 등 분석

Week 2 : Place & Route (PnR) Flow 구성

* RTL-to-GDS 플로우 정식 구축
* 각 단계별 명령 자동화 스크립트 구성
* Floorplan, Placement, Clock Tree Synthesis, Routing, DRC 진행
* and\_gate.v 기준으로 layout 확인 및 파형 검증

Week 3 : 툴 분석 및 커스터마이징

* OpenROAD 내부 Tcl 명령어 동작 방식 분석
* Floorplan 설정 변경 실험 (aspect ratio, track size 등)
* 로그 분석 자동화 및 PPA 추출 스크립트 일부 구현
* 사용자가 직접 .def, .spef, .gds 파일 생성 확인

Week 4 : 문서화 및 포트폴리오 정리

* 실습 결과를 Markdown 기반 문서로 정리
* 각 회로별로 다음 구조로 기록:
  + Verilog 설계
  + Testbench
  + Simulation 결과 이미지
  + Yosys 스크립트 / 합성 로그
  + OpenROAD 플로우 사용 명령
  + GDS 레이아웃 이미지

**5. 분석 및 이해 내용**

Yosys

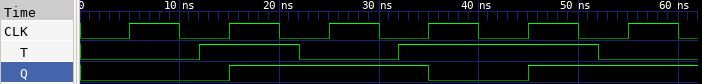
* RTL에서 게이트 수준 netlist를 생성하는 합성 도구
* read\_verilog, synth, write\_verilog 명령어로 사용
* Synthesis 결과로 .json 또는 .v netlist 출력

OpenROAD

* 완전한 자동 배치 및 배선 수행
* Tcl 기반 명령 스크립트를 통해 PnR 플로우 제어
* 단계:
  1. Floorplan: init\_floorplan.tcl → die/core 영역 설정
  2. Placement: 셀 위치 자동 배치
  3. CTS: 클럭 트리 생성
  4. Routing: 배선 처리
  5. DRC & GDS Export

**6. 실습 결과 이미지**

Verilog 회로 시뮬레이션 결과 (t\_flilp\_flop\_wave.png)



OpenROAD 배치 배선 결과

다채로움, 스크린샷, 패턴, 예술이(가) 표시된 사진

자동 생성된 설명

**7. 프로젝트를 통해 배운 점**

* 단순 사용자 관점에서 벗어나 툴의 내부 동작 방식에 대한 이해가 가능해졌음
* 실제 설계 플로우에 필요한 파일 포맷(DEF, LEF, GDS, Liberty 등)의 중요성 체감
* Tcl 스크립트 기반 자동화의 유연성과 어려움 동시에 경험
* 포트폴리오 구성 시 "회로 + 파형 + 합성 로그 + 배치 결과"가 일관되게 정리되어야 함을 인지

**8. GitHub 디렉토리 구조**

vlsi-lab/

├── step1-logic-gates/

├── step2-combinational/

├── step3-sequential/

├── step4-lib-analysis/

├── step5-openroad-flow-analysis/

**9. 결론**

이번 실습을 통해 RTL-to-GDSII 전 과정을 직접 실행해봄으로써, 단순한 회로 설계자 관점을 넘어 툴의 내부 구조와 제어 방식까지 이해할 수 있었다. 또한 오픈소스 툴을 분석하고 확장해보는 경험을 통해 실제 EDA Tool 개발자로 나아가기 위한 기초를 다질 수 있었다.